DOCUMENT RESUME

ED 248 861 IR 011 310

AUTHOR	Taitt, Kathy
TITLE	TLC for Growing Minds. APPLE II Programming Manual, Volume 5.
REPORT NO	ISBN-0-88193-005-9
PUB DATE	83
NOTE	60p.; For related documents, see IR 011 306-311, IR 011 314-316, and IR 011 318-319. This manual is also available for TRS-80, IBM PC, Commodore 64, PET, ATARI, TI-99/4A, and VIC-20.
AVAILABLE FROM	Creative Learning Association, Inc., RR#4, Box 330, Charleston, IL 61920 ($9.95 per copy; quantity discounts available).
PUB TYPE	Guides - Classroom Use - Materials (For Learner) (051)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Adult Education; *Computer Science Education; Computer Software; Elementary Secondary Education; Input Output Devices; Instructional Materials; *Learning Activities; *Microcomputers; Problem Solving; *Programing; Programing Languages
IDENTIFIERS	Apple II; *BASIC Programing Language

ABSTRACT

Designed to improve students' thinking, learning, and creative skills while they learn to program a microcomputer in BASIC programing language, this instructional manual for advanced learners provides a brief section of information for each new topic together with a sample program to enter, run on the computer, and observe; a suggested change to clarify the principle being taught, and a challenge requiring use of the new principles. Each chapter includes a review. The following commands are taught in this volume: GET N$, ASC(N$), LEFT$(X$,N), RIGHT$(X$,N), MID$(X$,J,N), LEN(N$), VAL(X$), STR$(X), ONERR GOTO, PEEK(X), and POKE X,Z. A reference chart displays commands and their functions. Separate color coded project books for junior high and high school/adult students are available for this manual. (Author/LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

THINKING · LEARNING · CREATING

TLC

for Growing Minds

APPLE II®

PROGRAMMING MANUAL

ED248861

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

✓ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Marilyn Burton

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR011310

Volume 5

THINKING · LEARNING · CREATING

TLC

APPLE II®
PROGRAMMING MANUAL

for Growing Minds

Volume 5

by Kathy Taitt

ion Resources

CREATIVE LEARNING ASSOCIATION, INC.
RR#4, Box 330
Charleston, Illinois 61920

ISBN 0-88193-005-9

© 1983 CREATIVE LEARNING ASSOCIATION, INC.

Charleston, Illinois 61920

All rights reserved. No part of this book, except for brief passages in articles and reviews that refer to both author and publisher, may be reproduced without written permission of the publisher.

Table of Contents

APPLE VOLUME V

CHAPTER 1: GET N$ AND ASC(N$)

Page

GET N$ 1

ASC(N$). 3

Review 10

CHAPTER 2: LEFT$, RIGHT$ AND MID$

LEFT$(X$,N). 11

RIGHT$(X$,N) 14

MID$(X$,J,N) 15

Review 18

CHAPTER 3: MORE STRINGS AND THINGS

LEN(N$). 19

VAL(X$). 22

STR$(X). 34

Review 38

CHAPTER 4: ONERR, PEEK AND POKE

ONERR GOTO 39

PEEK(X). 42

POKE X,Z 45

Review 52

Reference. 53

Creative Learning Association

Charleston, IL 61920

CHAPTER 1: GET N$ AND ASC(N$)

INFORMATION

The command, GET N$, allows you to enter a single character without pressing [RETURN].

ENTER/RUN/OBSERVE

```
10 HOME
20 GET N$
30 PRINT "YOU PRESSED ";N$;"."
40 GOTO 20
```

Now press a key. Press another one. (Press [CONTROL] [RESET] to stop the program.)

CHANGE IT/RUN/OBSERVE

```
22 GET M$
24 GET P$
30 PRINT N$;M$;P$
```

Press several keys one at a time including the space bar.

CHANGE IT/RUN/OBSERVE

Try this for coding a message.

```
15 PRINT "TYPE IN A SIX LETTER WORD"
3Ø PRINT P$;M$;N$;"   ";
```

CHALLENGE

Create a program that will code your typed message in groups of three letters. Can it also decode it if you type in the coded message?

INFORMATION

When a character is entered, ASC(N$) gives the ASC number that represents that character.

ASC stands for American Standard Code.

ENTER/RUN/OBSERVE

```
10 HOME
20 GET N$
30 PRINT ASC(N$),N$
40 GOTO 20
```

CHANGE IT/RUN/OBSERVE

Now make this change.

```
30 PRINT "THE ASC CODE FOR "N$" IS " ASC(N$)
```

INFORMATION

When you enter a string into the computer, the computer changes each symbol of the string into its corresponding ASC number. Your computer can alphabetize words using these numbers.

ENTER/RUN/OBSERVE

```
10 HOME
20 INPUT "ENTER A WORD ";W1$
30 INPUT "ENTER ANOTHER WORD ";W2$
40 PRINT:PRINT
50 IF W1$ < W2$ THEN PRINT W1$:PRINT W2$:
GOTO 70
60 PRINT W2$:PRINT W1$
70 FOR T=1 TO 2000:NEXT T
80 GOTO 10
```

The smaller the ASC number, the closer the letter is to the beginning of the alphabet. Line 50 compares the letters of the strings W1$ with those of W2$, one by one.

Creative Learning Association
Charleston, IL 61920

CHALLENGE

See if you can create a program that will PRINT three INPUTed words alphabetically. SAVE this program. Call it ALPHA.

If you are using a disk drive, be sure to load the DOS (boot up your system) before creating your program so you will be able to SAVE it.

INFORMATION

Computers are often used to code and decode messages.

ENTER/RUN/OBSERVE

To produce a secret number code, try this.

```
5 INPUT "ENTER A SMALL NUMBER "; N
10 GET L$
30 PRINT ASC(L$)+N
40 GOTO 10
```

CHANGE IT/RUN/OBSERVE

Or for a letter code, add these lines.

```
30 X=ASC(L$)+N
35 PRINT CHR$(X);
```

How does your friend know what number you added to each ASC code? Just assign each friend a single number code, such as 007.

To decode a message, try this.

```
**********************************************
*                                            *
*  3Ø X=ASC(L$)-N                            *
*                                            *
**********************************************
```

Now INPUT the code number and type in the message.

CHALLENGE

Create a program that will allow you to code or decode a message. SAVE this program. Call it CODE.

If you have a printer attached to your computer, print out a coded message to your friend.

INFORMATION

GET N$ can be used with many other commands. It is useful in games because the player doesn't have to press [RETURN] every time a character is entered.

ENTER/RUN/OBSERVE

```
10 GR
20 COLOR=2
30 X=20:Y=20
40 PLOT X,Y
50 GET G$
60 IF G$="I" THEN Y=Y-1
70 IF G$="M" THEN Y=Y+1
80 IF G$="J" THEN X=X-1
90 IF G$="K" THEN X=X+1
100 GOTO 40
```

Press the [I], [M], [J] and [K] keys and see what happens. If you have an APPLE IIe, try holding a key down for awhile. Try drawing a box. SAVE this program. Call it DRAW.

CHANGE IT/RUN/OBSERVE

```
 35 IF C=12 THEN C=Ø
 37 COLOR=2+C
 95 C=C+1
1ØØ GOTO 35
```

See what happens if you run the line off the screen.

CHANGE IT/RUN/OBSERVE

```
1ØØ IF X<Ø THEN X=39
11Ø IF X>39 THEN X=Ø
12Ø IF Y<Ø THEN Y=24
13Ø IF Y>24 THEN Y=Ø
14Ø GOTO 35
```

CHALLENGE

Change the program so that instead of drawing a line, you move a dot around on the screen.

REVIEW: Choose the command which causes the function described below.

COMMAND	FUNCTION
	This command returns the AMERICAN STANDARD CODE number of a given character.
	This command allows you to enter a single character without pressing RETURN.

CHAPTER 2: LEFT$, RIGHT$ AND MID$

INFORMATION

The command LEFT$(X$,N) tells the computer to look at the first N characters of the string X$, starting from the left.

ENTER/RUN/OBSERVE

```
10 HOME
20 X$="APPLE"
30 PRINT LEFT$(X$,1)
40 PRINT LEFT$(X$,3)
50 PRINT LEFT$(X$,5)
```

CHANGE IT/RUN/OBSERVE

Now make these changes.

```
20 X$="THIS IS HOW IT WORKS!"
30 PRINT LEFT$(X$,4)
40 PRINT LEFT$(X$,11)
50 PRINT LEFT$(X$,21)
```

RUN.

Now LIST. Do you see what happened?

CHANGE IT/RUN/OBSERVE

What if the number N is greater than the number of characters in the string? Add this line to find out what happens.

```
70 PRINT LEFT$(X$,30);"*"
```

Notice that the asterisk (*) was printed right after the last character of X$, the exclamation mark. LEFT$(X$,N) prints only the number of characters in the string. It doesn't print spaces if N is greater than the number of characters in the string.

CHANGE IT/RUN/OBSERVE

```
20 X$="FIREWORKS"
30 PRINT X$
40 PRINT LEFT$(X$,4)
50 PRINT LEFT$(X$,3)
```

CHALLENGE

Create a program using the word "WORKMANSHIP" and LEFT$ to select several words.

INFORMATION

LEFT$ is especially useful for analyzing INPUTed responses.

ENTER/RUN/OBSERVE

```
10 HOME
20 PRINT "DO YOU WANT TO GO AGAIN?"
30 INPUT A$
40 IF LEFT$(A$,1)="Y" THEN GOTO 10
50 END
```

With LEFT$ in the above program, you can handle answers of "YES", "Y", and "YEA".

CHANGE IT/RUN/OBSERVE

```
45 IF LEFT$(A$,1)="N" THEN 60
60 PRINT "QUITTER!"
```

CHALLENGE

Create your own program using LEFT$ that will give a different response when "YES" and "YEA" are used.

INFORMATION

RIGHT$(X$,N) is like LEFT$ except that the computer starts counting N from the right instead of the left.

ENTER/RUN/OBSERVE

```
10 HOME
20 X$="APPLE"
30 PRINT RIGHT$(X$,1)
40 PRINT RIGHT$(X$,3)
50 PRINT RIGHT$(X$,5)
```

Compare this program to the first program in the section on LEFT$. Notice the difference RIGHT$ makes.

CHANGE IT/RUN/OBSERVE

Add these lines.

```
25 PRINT LEFT$(X$,2);
35 END
```

CHALLENGE

Can you create a program that will print "SHIP WORKMAN" from WORKMANSHIP?

INFORMATION

MID$(X$,J,N) gives N characters of X$, starting at the Jth character from the left.

ENTER/RUN/OBSERVE

```
10 X$="ABCDEFGHI"
20 PRINT MID$(X$,4,1)
```

CHANGE IT/RUN/OBSERVE

```
30 PRINT MID$(X$,6,3)
```

CHANGE IT/RUN/OBSERVE

```
30 PRINT MID$(X$,9,1)
40 PRINT MID$(X$,3,1)
50 PRINT MID$(X$,5,1)
```

CHALLENGE

Using X$="FIREWORKS" and MID$(X$,J,N), see how many different words you can make.

INFORMATION

MID$ can be used to search for spaces--such as in the example below.

ENTER/RUN/OBSERVE

```
10 HOME:J=1
30 A$="PLEASE ENTER YOUR FIRST AND LAST
NAME."
40 PRINT A$
50 INPUT N$
60 IF MID$(N$,J,1)=" " GOTO 80
70 J=J+1:GOTO 60
80 HOME
90 PRINT "HELLO ";MID$(N$,J)
```

Notice what happens in line 90 when the number of characters, N, is not given. (The computer assumes N=1.)

CHANGE IT/RUN/OBSERVE

Now add this.

```
100 PRINT:PRINT "OR MAY I CALL YOU ";
110 PRINT MID$(N$,1,J);"?"
```

SAVE this program. Call it NAME.

ENTER/RUN/OBSERVE

Clear the memory with NEW, then enter this program.

```
100 A$="COUNT THE SPACES IN THIS MESSAGE."
120 K=1
130 IF MID$(A$,K,1)=" " GOTO 160
140 IF MID$(A$,K,1)="." GOTO 170
150 K=K+1:GOTO 130
160 Y=Y+1:K=K+1:GOTO 130
170 PRINT:PRINT:PRINT "THERE WERE ";Y;
" SPACES IN THE MESSAGE."
180 END
```

CHALLENGE

Create a program that will allow you to INPUT a sentence as a string L$ and then have the computer count the number of times the letter A was used. SAVE this program. Call it LETTER COUNT.

Creative Learning Association
Charleston, IL 61920

REVIEW: Choose the command which causes the function described below.

COMMAND	FUNCTION
	Use this to select a certain number of characters (N) of a string (X$), starting from the left.
	This command can be used to select characters starting from the right.
	This command selects a certain number of characters (N) of a string (X$), starting from a particular point in the string (J).

CHAPTER 3: MORE STRINGS AND THINGS

INFORMATION

The command LEN(N$) gives the number of characters in the string N$. (LEN stands for length.)

ENTER/RUN/OBSERVE

```
10 HOME
20 INPUT "ENTER A WORD ";W$
40 PRINT W$;" HAS ";LEN(W$);" LETTERS IN
IT."
50 PRINT
60 GOTO 20
```

CHANGE IT/RUN/OBSERVE

```
20 INPUT "ENTER A MESSAGE.";M$
30 X=LEN(M$)
40 PRINT "YOUR MESSAGE HAD ";X;
" CHARACTERS."
```

Try several messages.

CHANGE IT/RUN/OBSERVE

```
30

40 PRINT "YOUR MESSAGE HAD " LEN(M$)
" CHARACTERS."
```

LIST.

Now RUN.

CHANGE IT/RUN/OBSERVE

Now LOAD NAME (from page 16) and add these lines.

```
120 FOR T=1 TO 2000:NEXT T

150 A=J-1

160 B=LEN(N$)-J

170 PRINT "THERE ARE ";A;" LETTERS IN YOUR
FIRST NAME,"

180 PRINT "AND ";B;" LETTERS IN YOUR LAST
NAME."
```

CHALLENGE

Can you change the program to handle a middle name?

SUPER CHALLENGE

LOAD LETTER COUNT and then add the necessary lines to calculate what percent of the characters in L$ are A's.

Hint: To get the percent, find how many times A is used, then divide that number by LEN(L$). Multiply the answer by 100, and you will have the percent of characters that are A's.

Try this sample:

```
L$="ALL ARE ABLE"
```

Did you get 25 percent?

INFORMATION

VAL(X$) turns a string into a number.

ENTER/RUN/OBSERVE

```
10 HOME
20 X$="9"
30 PRINT X$;" PLAYERS ARE ON THE FIELD AT
ONE TIME."
40 Y=X$-1
50 PRINT:PRINT "THE BATTER SEES ";Y;" OF
THEM."
```

Notice what happens when you try to run this program? The string, "9", has a different meaning to the computer than the number 9. You cannot perform mathematical operations on strings.

CHANGE IT/RUN/OBSERVE

```
40 Y=VAL(X$)-1
```

Now notice what happens.

CHANGE IT/RUN/OBSERVE

Add these lines.

```
60 Z=VAL(X$)/3
70 PRINT:PRINT Z;" OF THE ";Y;" PLAYERS THE
BATTER SEES ARE OUTFIELDERS."
```

When you have finished with this program, clear it from memory.

ENTER/RUN/OBSERVE

Here is a program that will add the digits in your phone number.

```
 5 HOME
10 INPUT "ENTER YOUR PHONE NUMBER.";P$
20 FOR K=1 TO LEN(P$)
30 X$=MID$(P$,K,1)
40 S=S+VAL(X$)
50 NEXT K
60 PRINT S
```

Try these numbers.

217-345-1010

(217)345-1010

217/345-1010

Notice the results!

(Please don't ask why anyone would ever want to do this.)

CHANGE IT/RUN/OBSERVE

```
10 PRINT "ENTER THREE EXPENSES"
20 FOR K=1 TO 3
30 INPUT X$
```

Now INPUT these three values.

```
 4.55
18.00
 1.98
```

Try them again, but this time use dollar signs.

```
 $4.55
$18.00
 $1.98
```

CHANGE IT/RUN/OBSERVE

By adding these lines, even dollar signs can be accepted.

```
31 FOR D=1 TO LEN(X$)
32 N$=MID$(X$,D,1)
```

(program continued)

```
34 IF N$="$" GOTO 38
36 D$=D$+N$
38 NEXT D
40 S=S+VAL(D$)
46 D$=""
```

Notice that line 32 takes the string apart, character by character. Line 34 throws away any dollar signs, and line 36 puts the characters back together again to form a new string without a dollar sign. Line 36 does not add the values of the characters together.

CHANGE IT/RUN/OBSERVE

In place of lines 31 through 38, you could use this.

```
35 IF LEFT$(X$,1)="$" THEN D$=RIGHT$(X$,
LEN(X$)-1)
```

CHALLENGE

Create a program that will allow you to INPUT weights in pounds and add them for a total weight.

43#
21#
16.3#

SUPER CHALLENGE

Can you create a program that will allow you to INPUT answers like 12 ft. 4 in. and add them. If so, try these!

12 ft. 4 in.
8 ft. 2 in.
6 in.

Creative Learning Association
Charleston, IL 61920

INFORMATION

This is a popular game called HANGMAN.

ENTER/RUN/OBSERVE

```
 10 HOME:CLEAR 300:DIM A$(20)
 20 F$="":B$=""
 30 DATA SELECT,CERTAIN,ALPHABET, CATCHER,
 DOLLAR,XXX
 50 REM *** INSTRUCTIONS ***
 60 PRINT "THE OBJECT OF THIS GAME IS TO
 GUESS THE LETTERS IN A MYSTERY"
 65 PRINT "     WORD IN LESS THAN SEVEN
 GUESSES.":PRINT:PRINT
100 REM *** LOAD THE WORD ARRAY FROM THE
DATA BANK ***
110 READ W$
115 W$(Y)=W$
120 Y=Y+1
130 IF W$<>"XXX" GOTO 110
200 REM *** RANDOMLY PICK A WORD ***
210 X=INT((Y-1)*RND(0))
220 W$=W$(X)
230 L=LEN(W$)
240 FOR T=1 TO L
250 N$(T)=MID$(W$,T,1)
260 NEXT T
270 GOSUB 4000
2000 REM *** GUESS A LETTER ***
```

(program continued)

```
2002 PRINT "GUESS A LETTER IN THE WORD"
2005 C=1
2010 GET L$
2015 IF L$="" THEN GOTO 2010
2018 HOME
2020 FOR T=1 TO L
2030 IF L$=N$(T) THEN GOSUB 3000
2040 NEXT T
2050 IF C=1 THEN GOSUB 5000
2500 GOSUB 6000
2900 GOTO 2000
3000 REM *** FOUND A LETTER ***
3010 A$(T)=L$
3047 C=2
3050 RETURN
4000 REM *** FILL ARRAY WITH SPACES ***
4010 FOR T=1 TO L
4020 A$(T)="-"
4030 NEXT T
4040 RETURN
5000 REM *** WRONG LETTER ***
5010 Z=Z+1
5020 H$="HANGMAN"
5040 B$=B$+" "+L$
5060 RETURN
6000 REM *** PRINT RESULTS ***
6010 PRINT "LETTERS FOUND","WRONG LETTERS",
"GALLOWS RESULT"
```

(program continued)

```
6020 FOR T=1 TO L
6030 F$=F$+A$(T)
6040 NEXT T
6045 IF F$=W$ THEN GOSUB 8000
6050 PRINT F$,B$,LEFT$(H$,Z)
6053 IF Z=7 THEN GOTO 7000
6055 F$=""
6070 RETURN
7000 REM *** LOST THE GAME ***
7010 PRINT:PRINT:PRINT "SORRY, BUT YOU LOST"
7020 PRINT:PRINT "THE WORD WAS ";W$;"."
7200 END
8000 REM *** WON THE GAME ***
8010 HOME
8020 PRINT:PRINT,,"CONGRATULATIONS!"
8030 PRINT,,,"YOU WON!"
8100 END
```

SAVE this program. Call it HANGMAN.

RUN it several times. Correct any errors.

Notice the form for putting strings together in lines 5040 and 6030.

CHANGE IT/RUN/OBSERVE

Add more words to the program.

RUN several times to be sure they are accepted.

CHANGE IT/RUN/OBSERVE

Add the necessary lines so that at the end of a game you have the option of playing another game.

CHALLENGE

Add to the above program, the ability to keep score. The scoring should be obtained by taking the total number of wrong guesses divided by the number of games won. When your game is working, SAVE it.

INFORMATION

Most of us, at one time or another, have played a game where we try to make as many words as possible from the letters of one big word. This game is particularly popular around holidays. From the word CHRISTMAS, for example, you can spell MAT, MATH, HAT, AT, etc.

The program below lets you enter a word, and prints all the three-letter combinations of that word. By looking at the list and trying various arrangements of the letters in each combination, you can find <u>all</u> three-letter words that can be made from that word.

ENTER/RUN/OBSERVE

```
10 HOME
20 INPUT "ENTER A WORD ";W$
30 PRINT
40 PRINT "HERE ARE ALL OF THE THREE-LETTER
COMBINATIONS OF ";W$;":"
50 PRINT:PRINT
```

(program continued)

```
 60 A=2
 70 FOR T=1 TO LEN(W$)-2
 80 FOR R=A TO LEN(W$)-1
 90 FOR X=A+1 TO LEN(W$)
100 PRINT MID$(W$,T,1);MID$(W$,R,1);MID$(W$,
X,1),
110 NEXT X
120 A=A+1
130 NEXT R
140 A=T+2
150 NEXT T
```

Notice how the comma at the end of line 100 causes the words to be printed in columns.

CHANGE IT/RUN/OBSERVE

```
 40 PRINT "HERE ARE ALL OF THE FOUR-LETTER
 COMBINATIONS OF ";W$;":"
 65 FOR Q=1 TO LEN(W$)-3
 70 FOR T=A TO LEN(W$)-2
 80 FOR R=A+1 TO LEN(W$)-1
 90 FOR X=A+2 TO LEN(W$)
100 PRINT MID$(W$,Q,1);MID$(W$,T,1);MID$
(W$,R,1);MID$(W$,X,1),
140 A=T+1
160 A=Q+2
170 NEXT Q
```

CHALLENGE

Using the last two programs as two of the subroutines, write a program that will print all of the two-letter, three-letter, four-letter, and five-letter combinations of an INPUTed word W$.

Create the program so that someone can INPUT which set of combinations he wants to see, and using the ON-GOSUB command, the computer will branch to that subroutine and print all the possible letter combinations on the screen.

Creative Learning Association
Charleston, IL 61920

INFORMATION

STR$(X) turns the number X into the string "X".

ENTER/RUN/OBSERVE

```
10 HOME:PRINT
20 PRINT "YOU HAVE JUST BEEN HIRED FOR A
JOB."
30 INPUT "HOW MANY DOLLARS DO YOU EARN PER
HOUR? ";W
40 N=W*40
50 PRINT:PRINT "FOR A 40-HOUR WEEK, YOU
WILL EARN ";N
```

CHANGE IT/RUN/OBSERVE

Not too impressive so far. Suppose you decide to add a dollar sign to N to give it more meaning.

```
40 D=W*40
45 N="$"+D
```

RUN and notice the error message.

The problem in line 45 is that you aren't allowed to add strings to numbers. You may

only add numbers to numbers or strings to strings.

```
CHANGE 5T/RUN/OBSERVE

 45 N$="$"+STR$(D)

 50 PRINT:PRINT "FOR A 40-HOUR WEEK YOU
 WILL EARN ";N$
```

Notice how this solves the problem.

```
ENTER/RUN/OBSERVE
```

Here is another example.

```
 10 HOME

 20 INPUT "ENTER A SIX-DIGIT NUMBER. ";N

 30 X=INT(RND(1)*10)

 40 PRINT:PRINT "THE RANDOM NUMBER IS ";X

 50 N$=STR$(N)

 60 FOR T=1 TO LEN(N$)

 70 IF MID$(N$,T,1)=STR$(X) THEN C=C+1

 80 NEXT T

 90 PRINT:PRINT "THE NUMBER YOU ENTERED
 HAD ";C;" OF THE RANDOM NUMBERS IN IT."
```

Notice that by changing the number to a string, the string commands can be used to check each digit.

ENTER/RUN/OBSERVE

Now clear memory and enter this program.

```
 10 HOME
 40 DIM A$(20)
 50 INPUT "ENTER A WHOLE NUMBER LESS THAN
TEN THOUSAND ";N$
 52 REM *** ADD THE NUMBERS ***
 55 S=S+VAL(N$)
 60 L=LEN(N$)
 65 REM *** ADD SPACES TO ALIGN THE
NUMBERS ***
 70 FOR T=1 TO 6-L
 80 S$=S$+CHR$(32)
 90 NEXT T
 95 REM *** ADD A DOLLAR SIGN ***
100 X$="$"+S$+N$
110 A$(C)=X$
111 REM *** STORE IT IN AN ARRAY ***
112 C=C+1
115 REM *** STOP IT AFTER THREE NUMBERS
***
118 IF C=3 GOTO 200
119 S$=""
120 GOTO 50
130 PRINT
200 FOR F=0 TO C
210 PRINT A$(F)
```

(program continued)

```
220 NEXT F
230 PRINT "$";S
```

CHALLENGE

Change the above program to add two columns of money, keeping the decimal aligned in both columns. When your program works, INPUT these numbers.

Column #1	Column #2
12345.67	.01
1234.56	.12
123.45	1.23
12.34	12.34
1.23	123.45
.12	1234.56
.01	12345.67

Creative Learning Association
Charleston, IL 61920

REVIEW: Choose the command which causes the function described below.

COMMAND	FUNCTION
	This command gives the number of characters or length of a string.
	This command changes strings into numbers.
	This command changes numbers into strings.

CHAPTER 4: ONERR, PEEK AND POKE

INFORMATION

Normally, when you tell the computer to RUN a program and it encounters an error, the program stops running and an error message is printed. You can avoid having this happen by using the command ONERR GOTO. This command must be followed by a line number. When placed early in a program, it tells the computer to GOTO a particular line if an error occurs while the program is running.

ENTER/RUN/OBSERVE

```
10 HOME
20 PRINT "WITHOUT ONERR GOTO"
30 A=5
40 B$=A+"%"
```

Notice the error message.

CHANGE IT/RUN/OBSERVE

Now add these lines.

```
20 ONERR GOTO 50
50 PRINT "WITH ONERR GOTO"
```

If an error is encountered in a program before an ONERR GOTO statement is read, the program will stop and the error message will be printed.

ENTER/RUN/OBSERVE

LOAD your program named DRAW, or reenter these lines.

```
10 GR
20 COLOR=2
30 X=20:Y=20
40 PLOT X,Y
50 GET G$
60 IF G$="I" THEN Y=Y-1
70 IF G$="M" THEN Y=Y+1
80 IF G$="J" THEN X=X-1
90 IF G$="K" THEN X=X+1
100 GOTO 40
```

As you already know, when the line tries to go past the edge of the screen, you get an ILLEGAL QUANTITY ERROR IN 40 message. Games are a good place to use ONERR GOTO.

CHANGE IT/RUN/OBSERVE

```
5 ONERR GOTO 11Ø

11Ø PRINT "YOU WENT OFF THE SCREEN!"

12Ø GOTO 3Ø
```

CHALLENGE

See if ONERR GOTO can be used with SYNTAX ERRORS such as typing PRIMT instead of PRINT.

INFORMATION

Commands are stored in the computer's memory in the form of numbers. The command PEEK(X) looks into the computer's memory and sees what number is stored there.

ENTER/RUN/OBSERVE

```
10 PRINT PEEK(100)
```

This program looks into the computer's memory and prints the contents of location 100. The memory location in parenthesis can be from -65535 through 65535.

CHANGE IT/RUN/OBSERVE

```
 5 INPUT X
10 PRINT PEEK(X)
20 GOTO 5
```

Creative Learning Association
Charleston, IL 61920

ENTER/RUN/OBSERVE

```
10 X=PEEK(-16336)
20 GOTO 10
```

The memory location -16336 stores a command for sound, but the beep is so quick you need to put it in a loop to hear it. By changing the loop, you can create different types of sounds.

CHANGE IT/RUN/OBSERVE

```
 5 FOR A=1 TO 20
20 NEXT A
```

CHANGE IT/RUN/OBSERVE

```
 5 FOR A=1 TO 200
```

CHANGE IT/RUN/OBSERVE

```
15 PRINT
```

ERIC Clearinghouse on Information Resources
School of Education
Syracuse University
Syracuse, New York 13210
(315) 423-3640

CHANGE IT/RUN/OBSERVE

```
13 PRINT
18 PRINT
```

CHALLENGE

Using PEEK(-16336), create your own sound-producing program.

Creative Learning Association
Charleston, IL 61920

INFORMATION

The command POKE X,Z places the ASC number Z in the memory location X. The range for Z is Ø-255. The range for X is Ø up to the amount of memory your computer has. For 16K computers, the range is Ø-16384, for 32K, Ø-32768, and so on. (1K=1Ø24 bytes of memory.) There are fewer locations to POKE than to PEEK because much of the computer's memory contains information needed tc run the computer--information that shouldn't, and usually can't, be altered by POKE. However, you can still PEEK at these locations to see what is there.

POKE is especially useful for printing characters on the screen and changing the dimensions of the screen printout.

ENTER/RUN/OBSERVE

```
10 HOME
20 POKE 1024,193
```

CHANGE IT/RUN/OBSERVE

```
30 POKE 1025,194
40 POKE 1026,195
```

INFORMATION

The memory locations for your screen are numbers 1024-2039. There are 40 spaces on a line, so the POKE numbers for the first line on the screen are 1024-1063. You can print ASC characters on the screen by POKEing the appropriate numbers.

CHANGE IT/RUN/OBSERVE

```
10 HOME
20 FOR X=1024 TO 1063
30 POKE X,Y
40 Y=Y+1
50 NEXT X
```

Notice that the ASC characters 0 through 39 are printed in inverse.

CHANGE IT/RUN/OBSERVE

```
15 Y=192
```

When using POKE, the ASC numbers are printed in three sets: inverse, flash and normal. To see all the ASC characters and the order in which memory locations are assigned to the screen, enter the following program.

ENTER/RUN/OBSERVE

```
10 HOME
20 X=0:B=1024
30 POKE B,X
40 X=X+1
50 B=B+1
60 FOR T=1 TO 100:NEXT T
70 IF X < 256 GOTO 30
80 END
```

CHANGE IT/RUN/OBSERVE

```
15 GR
```

CHALLENGE

Write a program that prints your name on the screen without using the command PRINT.

INFORMATION

Your Apple screen can hold 24 lines of text, with 40 characters on a line. Using POKE, you can change the dimensions of the area in which the text is displayed.

POKE 32,L sets the left margin for the screen display. POKE 33,R sets the right margin. POKE 34,T sets the top margin, and POKE 35,B sets the bottom margin. Since there are 40 characters to a line and 24 lines on the screen maximum, the ranges for the variables L,R,T and B are:

L=0-39
R=1-40
T=0-23
B=1-24

ENTER/RUN/OBSERVE

```
10 HOME
20 PRINT "HERE ARE THE NUMBERS FROM 1
TO 1000:"
30 PRINT:PRINT:PRINT
40 POKE 34,6
```

(program continued)

```
50 FOR Z=1 TO 1000
60 FOR T=1 TO 50:NEXT T
70 PRINT Z; " ";
80 NEXT Z
```

Without line 40, the top line would scroll up off the screen as the program was run.

Ask for several LISTings of your program, and notice how the screen text is now confined to the dimension you set.

CHANGE IT/RUN/OBSERVE

To return to the normal setting, enter.

```
POKE 34,0
```

ENTER/RUN/OBSERVE

```
10 HOME:PRINT:PRINT
20 HTAB 5:PRINT "X"
50 POKE 32,4
60 POKE 34,5
70 PRINT:PRINT
80 PRINT X
```

(program continued)

```
*110 FOR T=1 TO 50:NEXT T                    *
*                                            *
*120 X=X+1                                    *
*                                            *
*130 GOTO 80                                  *
*                                            *
**********************************************
```

If you run this program more than once, press CTRL RESET and enter HOME between runnings to completely clear the screen.

CHALLENGE

Using POKE to set the dimensions of the printed display, create a table or chart whose headings do not disappear off the screen when the program is run.

REVIEW: Choose the command which causes the function described below.

COMMAND	FUNCTION
	This command causes the program to jump to a particular line number if an error occurs while the program is running.
	This command gives the contents of a specified memory location.
	This command stores a number (Z) into the memory location (X).

REFERENCE

COMMAND	FUNCTION
ASC(N$)	This command returns the AMERICAN STANDARD CODE number of a given character.
GET N$	This command allows you to enter a single character without pressing RETURN.
LEFT$(X$,N)	Use this to select a certain number of characters (N) of a string (X$), starting from the left.
LEN(N$)	This command gives the number of characters or length of a string.
MID$(X$,J,N)	This command selects a certain number of characters (N) of a string (X$), starting from a particular point in the string (J).
ONERR GOTO	This command causes the program to jump to a particular line number if an error occurs while the program is running.
PEEK(X)	This command gives the contents of a specified memory location.
POKE X,Z	This command stores a number (Z) into the memory location (X).

REFERENCE (continued)

COMMAND	FUNCTION
RIGHT$(X$,N)	This command can be used to select characters starting from the right.
STR$(X)	This command changes numbers into strings.
VAL(X$)	This command changes strings into numbers.

CONGRATULATIONS!!!

You have now finished the BLUE instructional level of TLC FOR GROWING MINDS and have a good working vocabulary of BASIC commands.

You are now ready to work the BLUE projects in the TLC Advanced Project Book. They will help you gain experience using the commands you have already learned. When you have finished the BLUE projects, Volume 6 (INDIGO) of TLC FOR GROWING MINDS will help you to continue learning.

Creative Learning Association
Charleston, IL 61920